

What is claimed is:

1. A universal method for building financial statements from any trial balance without any rigid framework to respect and regardless the accounts' number or the sequence of the accounts of the trial balance:
making the distinction between two types of balances (financial statement items and totals) appearing on a financial statement;
enabling the user to group the accounts into financial statement items simply and rapidly, by pointing, through the data structures and the display module;
enabling the user to group the financial statement items into totals simply and rapidly, by pointing, through the data structures and the display module, and considering that any total in a financial statement comes from balances of preceding lines;
enabling the report generator to show balances in parentheses not if it is a credit balance, or if it is of opposite direction of the whole financial statement, but only if it is of opposite direction of the financial statement item, by assigning the direction of the first account of a financial statement item to the financial statement item, considering that accounting amounts, debit and credit, are always accumulated, never subtracted, and adding these amounts according to the accounting equation to get the resulting balances;
organizing the accounting trial balance data in memory into a doubly linked data structure containing a LINKTRANS field;
using distinct lists of pointers (chartptr and poste_ptr) to link the data structure of the trial balance;
using a LINK vector to link transactions to accounts and to save the data of the financial statements;
allowing the reading of any trial balance printed on disk, instead of paper, by any accounting software;
organizing the financial statements in memory into a doubly linked data structure containing a linetype field;
using distinct lists of pointers (efptr and totalptr) to link the data structure of the financial statements;
allowing its integration in any accounting software;
allowing its integration in any word processor software, spreadsheet software, editing software, etc.;
allowing its use as a universal financial statements generator;
organizing accounting transactions in memory into one doubly linked data structure containing five different types of transactions identified by the journal field, and into a corresponding display structure line;
linking each transaction to the accounts of the trial balance with the linkchart and/or the linkbank field, initialized according to the LINK vector;
allowing the reading of accounting transactions listed on transactions reports simply printed on disk, instead of paper, by accounting software;

optimizing, when control accounts have been identified, the loading in of the transactions by representing the debit line and the credit line of a transaction on only one transaction element in the transaction data structure with the linkbank field and only one display line, to double the number of transactions which can be loaded in the central memory;

permitting through the data structures, the generator report, and the display module to display the detail of any balance appearing on the financial statement report up to its source, thru 3 different levels;

providing the first level of detail, being the detail of the financial statement item, that is, the accounts and their balances, grouped in the item, and incorporating this detail in the financial statement report;

providing a second level of detail, being the detail of an account balance selected on the first level, displaying the transactions affecting the account balance;

providing a third level of detail, being the detail of a document selected on the second level, displaying all the transactions which compose this document, since a document can affect more than one account, the invention allowing this multi-allocation;

allowing dynamically the memory space for each element of each data structure, giving the capacity to load a number of transactions according to the capacity of the central memory of the computer used;

allowing to save on external memory unloaded transactions for further loading in;

allowing to save on, and to reload from, external memory, the content of the data structures.

2. An optimal method to organize accounting data in the central memory of a computer:

using a doubly linked data structure containing a LINKTRANS field for the accounting trial balance data;

using a distinct list of pointers to link the data structure element of the trial balance;

keeping a direction field in the trial balance data structure to determine the debit or credit sign of the account for the financial statement report and to identify control accounts for the optimization process of the transaction structure;

using a doubly linked data structure containing a linetype field for the financial statements data;

using a distinct list of pointers to link the financial statements data structure element;

using sub-lists of trial balance structure element pointers to group the accounts into financial statement items;

using sub-lists of financial statement structure element pointers to group the financial statement items into totals;

using one doubly linked data structure for the accounting transactions containing different types of transactions identified by the journal field;

building rapidly these reports for any selected period of date comparing the Julian date field of the transaction structure data;
 modifying easily the display lines according to the transactions report request and reinitializing it rapidly using the transaction data structure;
 building the distinct lists of pointers to provide sorted presentation without the need to perform any sorting;
 building the distinct lists of pointers to group the transactions by allocation account without the need to perform any sorting, where the transaction structure has been optimized;
 building the distinct lists of pointers to combine inflow and outflow transactions in one report to use the reconciliation module, or any other transactions report;
 inserting a preinitialized display line to the linked list to get, in a simple way, a formfeed at the printing;
 using a new universal algorithm to add page headers and formfeed to any report;
 permitting the display of the reports, with the display module, before printing it.

4. A new universal algorithm to add page headers and formfeed to any report: incrementing the current element pointer before testing the line counter.
5. A method for displaying any data structure doubly linked by a distinct list of pointers, and for manipulating this display by modifying directly the video RAM (not using existing subroutines):
 - manipulating from top to bottom or bottom to top;
 - permitting partial display of the data structure lines, that is, the number of characters to be displayed can vary and the first character displayed can be different from the first character of the data structure line;
 - manipulating this partial display from right to left and left to right;
 - performing any change of one line (or one column) of the display working within the video RAM, making these changes very fast;
 - enabling the user to select an element;
 - enabling the user to do successive selection;
 - enabling the user to select screen color with the color selection module using one rainbow of colors for the foreground and one for the background or, using a matrix of all possible choices and modifying the byte for the character attribute in the video RAM;
 - enabling the user to get a calendar while consulting the display;
 - enabling the user to get a calculator while consulting the display;
 - enabling the user to get a tax calculation module while consulting the display.
6. A method for printing reports on any printer:
 - using a conversion vector to convert the ASCII code of all characters to print within ASCII codes 32 to 126, standard to all printers;
 - eliminating the need for an installation step for the printer;
 - enabling the user to initialize this conversion vector for specific printers;
 - permitting partial printing according to a partial display.
7. A universal accounting reconciliation method which, with its calculation unit, displays a theoretical balance throughout the process of checking the

transactions, avoiding the need to print the reconciliation report to know if the balance of the account in the books has been reconciled successfully and avoiding the need to go back to the checking process if not.

8. An insertion algorithm adding a new element to a doubly linked data structure.
9. A destruction algorithm removing an element to a doubly linked data structure.
10. A method to get dates from the data entry screen:
 - displaying a default date;
 - enabling the user to modify each portion of an input date field with the keys plus (+) or minus (-) or with numbers;
 - limiting the change of the year with the keys plus (+) or minus (-) in cases where the year portion of the date does not include the millennium, to assure the change to the right millennium;
 - enabling the user to get a calendar by pressing the key '.', which is the only key on the numerical keyboard not used in the input date field, and to select a date to transfer in the input field date.
11. A method to get amounts from the data entry screen:
 - grouping the numbers by thousands;
 - enabling the user to get a calculator by pressing the key '+' and to transfer the result of calculations in the input amount field;
 - enabling the user to get a tax calculation module by pressing the key '*' and to transfer the results of calculations in the input amount field.